[0019] Automatic left vs. right hand differentiation;

[0020] Input surface sub-region contact rejection (i.e., palm-rejection) coupled with any of the aforementioned examples; and

[0021] Differentiation of palm rejection algorithm based on type of computing device (e.g., e-reader vs. slate vs. dual-screen booklet vs. horizontal surface vs. wall-mounted display, etc.) as well as the specific sensing properties and capabilities of the touch, stylus, and other input modalities.

[0022] In view of the above summary, it is clear that the Contact Discriminator described herein provides various techniques for differentiating between intentional user contacts and unintentional user contacts on contact-sensitive computing devices. In addition to the just described benefits, other advantages of the Contact Discriminator will become apparent from the detailed description that follows hereinafter when taken in conjunction with the accompanying drawings.

DESCRIPTION OF THE DRAWINGS

[0023] The specific features, aspects, and advantages of the claimed subject matter will become better understood with regard to the following description, appended claims, and accompanying drawings where:

[0024] FIG. **1** illustrates an exemplary touch-sensitive computing device capable of being controlled by a "Contact Discriminator", as described herein.

[0025] FIG. **2** illustrates an example of using a "Contact Discriminator" to differentiate between intended touch inputs and unintended touch inputs by a user holding a touch-sensitive computing device in one hand while providing a stylus-based input to that device via a pen or stylus held in the other hand, as described herein.

[0026] FIG. **3** illustrates an exemplary architectural flow diagram that illustrates program modules for differentiating between intentional user contacts and unintentional user contacts on one or more contact-sensitive surfaces of a computing device using the "Contact Discriminator," as described herein.

[0027] FIG. **4** illustrates an example of using the "Contact Discriminator" to disable sub-regions of a touch-sensitive surface, or reject input originating within such sub-regions, resulting from contacts of portions of a user's hand or palm while concurrently accepting or allowing separate pen-touch and finger-touch type inputs, as described herein.

[0028] FIG. **5** illustrates an example of using the "Contact Discriminator" to disable sub-regions of a touch-sensitive surface, or reject input originating within such sub-regions, resulting from contacts of portions of a user's hand or palm while concurrently accepting or allowing concurrent finger-touch type inputs with that same hand, as described herein.

[0029] FIG. **6** illustrates an example of using the "Contact Discriminator" to determine relative angles or orientation between one or more pen or stylus type input devices and touch sensitive surfaces as those pen or stylus type devices hover over or approach the touch sensitive surfaces, as described herein.

[0030] FIG. **7** is a general system diagram depicting a simplified general-purpose computing device having simplified computing and I/O capabilities for use in implementing various embodiments of the Contact Discriminator, as described herein.

DETAILED DESCRIPTION OF THE EMBODIMENTS

[0031] In the following description of the embodiments of the claimed subject matter, reference is made to the accompanying drawings, which form a part hereof, and in which is shown by way of illustration specific embodiments in which the claimed subject matter may be practiced. It should be understood that other embodiments may be utilized and structural changes may be made without departing from the scope of the presently claimed subject matter.

[0032] 1.0 Introduction:

[0033] In general, a "Contact Discriminator," as described herein, provides various techniques for differentiating between valid and invalid contacts received from any input methodology by one or more touch-sensitive surfaces of a touch-sensitive computing device. Examples of contacts include single, sequential, concurrent, or simultaneous user finger touches (including gesture type touches), pen or stylus touches or inputs, hover-type inputs, or any combination thereof. The Contact Discriminator then acts on valid contacts (i.e., contacts intended as inputs) while rejecting or ignoring invalid contacts or inputs.

[0034] Advantageously, the Contact Discriminator is further capable of disabling or ignoring one or more regions or sub-regions of input surfaces that are expected to receive unintentional contacts, or intentional contacts not intended as inputs, for device or application control purposes. Examples of contacts not intended as inputs include, but are not limited to, a user's palm resting on a touch screen while the user writes on that screen with a stylus or pen.

[0035] Note that disabled or rejected sub-regions of the input surface are also referred to herein using the term "palm-rejection" region. Note further that in some embodiments "disabled" regions may still report input, but said input will be appropriately tagged (or its interpretation altered) so as to fully and properly account for the likelihood that it may represent unintended input. Hence, such altered interpretation might still employ such input as contextual data or in support of other application or system behaviors or feedback that differ, perhaps even quite subtly, from normal system operation without "palm rejection." Furthermore such responses may be gradated based on the probability that said input represents an incidental input, such that the system concurrently propagates both "intended" and "unintended" interpretations of an input during periods of uncertainty. Such propagated inputs can then be reversed or undone, if necessary, if the Contact Discriminator makes a final determination as to the validity of such contacts. Note also that the concept of a "latency period" as discussed in further detail herein is used in various embodiments to enable these concepts.

[0036] The Contact Discriminator is adaptable for use with any touch-sensitive computing device having one or more touch-sensitive surfaces or regions (e.g., touch screen, touch sensitive bezel or case, sensors for detection of hover-type inputs, optical touch sensors, etc.). Examples of touch-based computing devices include, but are not limited to, touch-sensitive display devices connected to a computing device, touch-sensitive phone devices, touch-sensitive media players, touch-sensitive e-reader, notebook, netbook, booklet (dual-screen), or tablet type computers, or any other device